Plagiarism Detection Model

Now that you've created training and test data, you are ready to define and train a model. Your goal in this notebook, will be to train a binary classification model that learns to label an answer file as either plagiarized or not, based on the features you provide the model.

This task will be broken down into a few discrete steps:

- **Upload your data to S3.**
- **Define a binary classification model and a training script.**
- **Train your model and deploy it.**
- **Evaluate your deployed classifier and answer some questions about your approach.**

To complete this notebook, you'll have to complete all given exercises and answer all the questions in this notebook.

> **All your tasks will be clearly labeled EXERCISE and questions as QUESTION.**

It will be up to you to explore different classification models and decide on a model that gives you the best performance for this dataset.

Load Data to S3

In the last notebook, you should have created two files: a `training.csv` and `test.csv` file with the features and class labels for the given corpus of plagiarized/non-plagiarized text data.

> **The below cells load in some AWS SageMaker libraries and creates a default bucket. After creating this bucket, you can upload your locally stored data to S3.**

Save your train and test `.csv` feature files, locally. To do this you can run the second notebook "2_Plagiarism_Feature_Engineering" in SageMaker or you can manually upload your files to this notebook using the upload icon in Jupyter Lab. Then you can upload local files to S3 by using `sagemaker_session.upload_data` and pointing directly to where the training data is saved.

In [1]:

```
import pandas as pd
import boto3
import sagemaker

!pip install sagemaker
```

```
Requirement already satisfied: sagemaker in /home/ec2-user/anaconda3/envs/pytorch_p36/lib
/python3.6/site-packages (2.72.1)
Requirement already satisfied: packaging>=20.0 in /home/ec2-user/anaconda3/envs/pytorch_p
36/lib/python3.6/site-packages (from sagemaker) (21.3)
Requirement already satisfied: importlib-metadata>=1.4.0 in /home/ec2-user/anaconda3/envs
/pytorch_p36/lib/python3.6/site-packages (from sagemaker) (4.5.0)
Requirement already satisfied: google-pasta in /home/ec2-user/anaconda3/envs/pytorch_p36/
lib/python3.6/site-packages (from sagemaker) (0.2.0)
Requirement already satisfied: pandas in /home/ec2-user/anaconda3/envs/pytorch_p36/lib/py
thon3.6/site-packages (from sagemaker) (1.1.5)
Requirement already satisfied: boto3>=1.20.18 in /home/ec2-user/anaconda3/envs/pytorch_p3
6/lib/python3.6/site-packages (from sagemaker) (1.20.25)
Requirement already satisfied: numpy>=1.9.0 in /home/ec2-user/anaconda3/envs/pytorch_p36/
lib/python3.6/site-packages (from sagemaker) (1.19.5)
Requirement already satisfied: smdebug-rulesconfig==1.0.1 in /home/ec2-user/anaconda3/env
s/pytorch_p36/lib/python3.6/site-packages (from sagemaker) (1.0.1)
Requirement already satisfied: attrs in /home/ec2-user/anaconda3/envs/pytorch_p36/lib/pyt
```

```
hon3.6/site-packages (from sagemaker) (21.2.0)
Requirement already satisfied: protobuf>=3.1 in /home/ec2-user/anaconda3/envs/pytorch_p36
/lib/python3.6/site-packages (from sagemaker) (3.17.2)
Requirement already satisfied: pathos in /home/ec2-user/anaconda3/envs/pytorch_p36/lib/py
thon3.6/site-packages (from sagemaker) (0.2.8)
Requirement already satisfied: protobuf3-to-dict>=0.1.5 in /home/ec2-user/anaconda3/envs/
pytorch_p36/lib/python3.6/site-packages (from sagemaker) (0.1.5)
Requirement already satisfied: jmespath<1.0.0,>=0.7.1 in /home/ec2-user/anaconda3/envs/py
torch_p36/lib/python3.6/site-packages (from boto3>=1.20.18->sagemaker) (0.10.0)
Requirement already satisfied: botocore<1.24.0,>=1.23.25 in /home/ec2-user/anaconda3/envs
/pytorch_p36/lib/python3.6/site-packages (from boto3>=1.20.18->sagemaker) (1.23.25)
Requirement already satisfied: s3transfer<0.6.0,>=0.5.0 in /home/ec2-user/anaconda3/envs/
pytorch_p36/lib/python3.6/site-packages (from boto3>=1.20.18->sagemaker) (0.5.0)
Requirement already satisfied: typing-extensions>=3.6.4 in /home/ec2-user/anaconda3/envs/
pytorch_p36/lib/python3.6/site-packages (from importlib-metadata>=1.4.0->sagemaker) (3.10
.0.0)
Requirement already satisfied: zipp>=0.5 in /home/ec2-user/anaconda3/envs/pytorch_p36/lib
/python3.6/site-packages (from importlib-metadata>=1.4.0->sagemaker) (3.4.1)
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in /home/ec2-user/anaconda3/envs/
pytorch_p36/lib/python3.6/site-packages (from packaging>=20.0->sagemaker) (2.4.7)
Requirement already satisfied: six>=1.9 in /home/ec2-user/anaconda3/envs/pytorch_p36/lib/
python3.6/site-packages (from protobuf>=3.1->sagemaker) (1.16.0)
Requirement already satisfied: python-dateutil>=2.7.3 in /home/ec2-user/anaconda3/envs/py
torch_p36/lib/python3.6/site-packages (from pandas->sagemaker) (2.8.1)
Requirement already satisfied: pytz>=2017.2 in /home/ec2-user/anaconda3/envs/pytorch_p36/
lib/python3.6/site-packages (from pandas->sagemaker) (2021.1)
Requirement already satisfied: dill>=0.3.4 in /home/ec2-user/anaconda3/envs/pytorch_p36/l
ib/python3.6/site-packages (from pathos->sagemaker) (0.3.4)
Requirement already satisfied: ppft>=1.6.6.4 in /home/ec2-user/anaconda3/envs/pytorch_p36
/lib/python3.6/site-packages (from pathos->sagemaker) (1.6.6.4)
Requirement already satisfied: pox>=0.3.0 in /home/ec2-user/anaconda3/envs/pytorch_p36/li
b/python3.6/site-packages (from pathos->sagemaker) (0.3.0)
Requirement already satisfied: multiprocess>=0.70.12 in /home/ec2-user/anaconda3/envs/pyt
orch_p36/lib/python3.6/site-packages (from pathos->sagemaker) (0.70.12.2)
Requirement already satisfied: urllib3<1.27,>=1.25.4 in /home/ec2-user/anaconda3/envs/pyt
orch_p36/lib/python3.6/site-packages (from botocore<1.24.0,>=1.23.25->boto3>=1.20.18->sag
emaker) (1.26.5)
```

In [2]:

```
"""
DON'T MODIFY ANYTHING IN THIS CELL THAT IS BELOW THIS LINE
"""
# session and role
sagemaker_session = sagemaker.Session()
role = sagemaker.get_execution_role()

# create an S3 bucket
bucket = sagemaker_session.default_bucket()
```

EXERCISE: Upload your training data to S3

Specify the `data_dir` where you've saved your `train.csv` file. Decide on a descriptive `prefix` that defines where your data will be uploaded in the default S3 bucket. Finally, create a pointer to your training data by calling `sagemaker_session.upload_data` and passing in the required parameters. It may help to look at the Session documentation or previous SageMaker code examples.

You are expected to upload your entire directory. Later, the training script will only access the `train.csv` file.

In [3]:

```
# should be the name of directory you created to save your features data
data_dir = 'plagiarism_data'

# set prefix, a descriptive name for a directory
prefix = 'plagiarism_detection'

# upload all data to S3
```

```
session = sagemaker.Session() # Store the current SageMaker session
input_data = session.upload_data(data_dir, bucket = bucket, key_prefix=prefix)
```

Test cell

Test that your data has been successfully uploaded. The below cell prints out the items in your S3 bucket and will throw an error if it is empty. You should see the contents of your `data_dir` and perhaps some checkpoints. If you see any other files listed, then you may have some old model files that you can delete via the S3 console (though, additional files shouldn't affect the performance of model developed in this notebook).

In [4]:

```
"""
DON'T MODIFY ANYTHING IN THIS CELL THAT IS BELOW THIS LINE
"""
# confirm that data is in S3 bucket
empty_check = []
for obj in boto3.resource('s3').Bucket(bucket).objects.all():
    empty_check.append(obj.key)
    print(obj.key)

assert len(empty_check) !=0, 'S3 bucket is empty.'
print('Test passed!')
```

```
plagiarism_detection/test.csv
plagiarism_detection/train.csv
sagemaker-scikit-learn-2022-01-04-13-02-27-134/source/sourcedir.tar.gz
sagemaker-scikit-learn-2022-01-04-13-03-15-684/source/sourcedir.tar.gz
sagemaker-scikit-learn-2022-01-04-13-15-49-152/debug-output/training_job_end.ts
sagemaker-scikit-learn-2022-01-04-13-15-49-152/output/model.tar.gz
sagemaker-scikit-learn-2022-01-04-13-15-49-152/source/sourcedir.tar.gz
sagemaker-scikit-learn-2022-01-04-13-40-36-023/source/sourcedir.tar.gz
sagemaker-scikit-learn-2022-01-06-10-40-24-087/profiler-output/system/incremental/2022010
610/1641465780.algo-1.json
sagemaker-scikit-learn-2022-01-06-10-40-24-087/profiler-output/system/incremental/2022010
610/1641465840.algo-1.json
sagemaker-scikit-learn-2022-01-06-10-40-24-087/source/sourcedir.tar.gz
sagemaker-scikit-learn-2022-01-06-10-46-48-701/profiler-output/system/incremental/2022010
610/1641466140.algo-1.json
sagemaker-scikit-learn-2022-01-06-10-46-48-701/profiler-output/system/incremental/2022010
610/1641466200.algo-1.json
sagemaker-scikit-learn-2022-01-06-10-46-48-701/rule-output/ProfilerReport-1641466008/prof
iler-output/profiler-report.html
sagemaker-scikit-learn-2022-01-06-10-46-48-701/rule-output/ProfilerReport-1641466008/prof
iler-output/profiler-report.ipynb
sagemaker-scikit-learn-2022-01-06-10-46-48-701/rule-output/ProfilerReport-1641466008/prof
iler-output/profiler-reports/BatchSize.json
sagemaker-scikit-learn-2022-01-06-10-46-48-701/rule-output/ProfilerReport-1641466008/prof
iler-output/profiler-reports/CPUBottleneck.json
sagemaker-scikit-learn-2022-01-06-10-46-48-701/rule-output/ProfilerReport-1641466008/prof
iler-output/profiler-reports/Dataloader.json
sagemaker-scikit-learn-2022-01-06-10-46-48-701/rule-output/ProfilerReport-1641466008/prof
iler-output/profiler-reports/GPUMemoryIncrease.json
sagemaker-scikit-learn-2022-01-06-10-46-48-701/rule-output/ProfilerReport-1641466008/prof
iler-output/profiler-reports/IOBottleneck.json
sagemaker-scikit-learn-2022-01-06-10-46-48-701/rule-output/ProfilerReport-1641466008/prof
iler-output/profiler-reports/LoadBalancing.json
sagemaker-scikit-learn-2022-01-06-10-46-48-701/rule-output/ProfilerReport-1641466008/prof
iler-output/profiler-reports/LowGPUUtilization.json
sagemaker-scikit-learn-2022-01-06-10-46-48-701/rule-output/ProfilerReport-1641466008/prof
iler-output/profiler-reports/MaxInitializationTime.json
sagemaker-scikit-learn-2022-01-06-10-46-48-701/rule-output/ProfilerReport-1641466008/prof
iler-output/profiler-reports/OverallFrameworkMetrics.json
sagemaker-scikit-learn-2022-01-06-10-46-48-701/rule-output/ProfilerReport-1641466008/prof
iler-output/profiler-reports/OverallSystemUsage.json
sagemaker-scikit-learn-2022-01-06-10-46-48-701/rule-output/ProfilerReport-1641466008/prof
iler-output/profiler-reports/StepOutlier.json
sagemaker-scikit-learn-2022-01-06-10-46-48-701/source/sourcedir.tar.gz
sagemaker-scikit-learn-2022-01-06-10-56-03-345/profiler-output/system/incremental/2022010
610/1641466680.algo-1.json
```

sagemaker-scikit-learn-2022-01-06-10-56-03-345/profiler-output/system/incremental/2022010
610/1641466740.algo-1.json
sagemaker-scikit-learn-2022-01-06-10-56-03-345/profiler-output/system/incremental/2022010
611/1641466800.algo-1.json
sagemaker-scikit-learn-2022-01-06-10-56-03-345/rule-output/ProfilerReport-1641466563/prof
iler-output/profiler-report.html
sagemaker-scikit-learn-2022-01-06-10-56-03-345/rule-output/ProfilerReport-1641466563/prof
iler-output/profiler-report.ipynb
sagemaker-scikit-learn-2022-01-06-10-56-03-345/rule-output/ProfilerReport-1641466563/prof
iler-output/profiler-reports/BatchSize.json
sagemaker-scikit-learn-2022-01-06-10-56-03-345/rule-output/ProfilerReport-1641466563/prof
iler-output/profiler-reports/CPUBottleneck.json
sagemaker-scikit-learn-2022-01-06-10-56-03-345/rule-output/ProfilerReport-1641466563/prof
iler-output/profiler-reports/Dataloader.json
sagemaker-scikit-learn-2022-01-06-10-56-03-345/rule-output/ProfilerReport-1641466563/prof
iler-output/profiler-reports/GPUMemoryIncrease.json
sagemaker-scikit-learn-2022-01-06-10-56-03-345/rule-output/ProfilerReport-1641466563/prof
iler-output/profiler-reports/IOBottleneck.json
sagemaker-scikit-learn-2022-01-06-10-56-03-345/rule-output/ProfilerReport-1641466563/prof
iler-output/profiler-reports/LoadBalancing.json
sagemaker-scikit-learn-2022-01-06-10-56-03-345/rule-output/ProfilerReport-1641466563/prof
iler-output/profiler-reports/LowGPUUtilization.json
sagemaker-scikit-learn-2022-01-06-10-56-03-345/rule-output/ProfilerReport-1641466563/prof
iler-output/profiler-reports/MaxInitializationTime.json
sagemaker-scikit-learn-2022-01-06-10-56-03-345/rule-output/ProfilerReport-1641466563/prof
iler-output/profiler-reports/OverallFrameworkMetrics.json
sagemaker-scikit-learn-2022-01-06-10-56-03-345/rule-output/ProfilerReport-1641466563/prof
iler-output/profiler-reports/OverallSystemUsage.json
sagemaker-scikit-learn-2022-01-06-10-56-03-345/rule-output/ProfilerReport-1641466563/prof
iler-output/profiler-reports/StepOutlier.json
sagemaker-scikit-learn-2022-01-06-10-56-03-345/source/sourcedir.tar.gz
sagemaker-scikit-learn-2022-01-06-11-12-03-909/profiler-output/system/incremental/2022010
611/1641467640.algo-1.json
sagemaker-scikit-learn-2022-01-06-11-12-03-909/profiler-output/system/incremental/2022010
611/1641467700.algo-1.json
sagemaker-scikit-learn-2022-01-06-11-12-03-909/rule-output/ProfilerReport-1641467523/prof
iler-output/profiler-report.html
sagemaker-scikit-learn-2022-01-06-11-12-03-909/rule-output/ProfilerReport-1641467523/prof
iler-output/profiler-report.ipynb
sagemaker-scikit-learn-2022-01-06-11-12-03-909/rule-output/ProfilerReport-1641467523/prof
iler-output/profiler-reports/BatchSize.json
sagemaker-scikit-learn-2022-01-06-11-12-03-909/rule-output/ProfilerReport-1641467523/prof
iler-output/profiler-reports/CPUBottleneck.json
sagemaker-scikit-learn-2022-01-06-11-12-03-909/rule-output/ProfilerReport-1641467523/prof
iler-output/profiler-reports/Dataloader.json
sagemaker-scikit-learn-2022-01-06-11-12-03-909/rule-output/ProfilerReport-1641467523/prof
iler-output/profiler-reports/GPUMemoryIncrease.json
sagemaker-scikit-learn-2022-01-06-11-12-03-909/rule-output/ProfilerReport-1641467523/prof
iler-output/profiler-reports/IOBottleneck.json
sagemaker-scikit-learn-2022-01-06-11-12-03-909/rule-output/ProfilerReport-1641467523/prof
iler-output/profiler-reports/LoadBalancing.json
sagemaker-scikit-learn-2022-01-06-11-12-03-909/rule-output/ProfilerReport-1641467523/prof
iler-output/profiler-reports/LowGPUUtilization.json
sagemaker-scikit-learn-2022-01-06-11-12-03-909/rule-output/ProfilerReport-1641467523/prof
iler-output/profiler-reports/MaxInitializationTime.json
sagemaker-scikit-learn-2022-01-06-11-12-03-909/rule-output/ProfilerReport-1641467523/prof
iler-output/profiler-reports/OverallFrameworkMetrics.json
sagemaker-scikit-learn-2022-01-06-11-12-03-909/rule-output/ProfilerReport-1641467523/prof
iler-output/profiler-reports/OverallSystemUsage.json
sagemaker-scikit-learn-2022-01-06-11-12-03-909/rule-output/ProfilerReport-1641467523/prof
iler-output/profiler-reports/StepOutlier.json
sagemaker-scikit-learn-2022-01-06-11-12-03-909/source/sourcedir.tar.gz
sagemaker-scikit-learn-2022-01-06-11-28-17-700/debug-output/training_job_end.ts
sagemaker-scikit-learn-2022-01-06-11-28-17-700/output/model.tar.gz
sagemaker-scikit-learn-2022-01-06-11-28-17-700/profiler-output/framework/training_job_end
.ts
sagemaker-scikit-learn-2022-01-06-11-28-17-700/profiler-output/system/incremental/2022010
611/1641468660.algo-1.json
sagemaker-scikit-learn-2022-01-06-11-28-17-700/profiler-output/system/incremental/2022010
611/1641468720.algo-1.json
sagemaker-scikit-learn-2022-01-06-11-28-17-700/profiler-output/system/training_job_end.ts
sagemaker-scikit-learn-2022-01-06-11-28-17-700/rule-output/ProfilerReport-1641468497/prof

```
iler-output/profiler-report.html
sagemaker-scikit-learn-2022-01-06-11-28-17-700/rule-output/ProfilerReport-1641468497/prof
iler-output/profiler-report.ipynb
sagemaker-scikit-learn-2022-01-06-11-28-17-700/rule-output/ProfilerReport-1641468497/prof
iler-output/profiler-reports/BatchSize.json
sagemaker-scikit-learn-2022-01-06-11-28-17-700/rule-output/ProfilerReport-1641468497/prof
iler-output/profiler-reports/CPUBottleneck.json
sagemaker-scikit-learn-2022-01-06-11-28-17-700/rule-output/ProfilerReport-1641468497/prof
iler-output/profiler-reports/Dataloader.json
sagemaker-scikit-learn-2022-01-06-11-28-17-700/rule-output/ProfilerReport-1641468497/prof
iler-output/profiler-reports/GPUMemoryIncrease.json
sagemaker-scikit-learn-2022-01-06-11-28-17-700/rule-output/ProfilerReport-1641468497/prof
iler-output/profiler-reports/IOBottleneck.json
sagemaker-scikit-learn-2022-01-06-11-28-17-700/rule-output/ProfilerReport-1641468497/prof
iler-output/profiler-reports/LoadBalancing.json
sagemaker-scikit-learn-2022-01-06-11-28-17-700/rule-output/ProfilerReport-1641468497/prof
iler-output/profiler-reports/LowGPUUtilization.json
sagemaker-scikit-learn-2022-01-06-11-28-17-700/rule-output/ProfilerReport-1641468497/prof
iler-output/profiler-reports/MaxInitializationTime.json
sagemaker-scikit-learn-2022-01-06-11-28-17-700/rule-output/ProfilerReport-1641468497/prof
iler-output/profiler-reports/OverallFrameworkMetrics.json
sagemaker-scikit-learn-2022-01-06-11-28-17-700/rule-output/ProfilerReport-1641468497/prof
iler-output/profiler-reports/OverallSystemUsage.json
sagemaker-scikit-learn-2022-01-06-11-28-17-700/rule-output/ProfilerReport-1641468497/prof
iler-output/profiler-reports/StepOutlier.json
sagemaker-scikit-learn-2022-01-06-11-28-17-700/source/sourcedir.tar.gz
sagemaker-scikit-learn-2022-01-06-13-32-04-990/profiler-output/system/incremental/2022010
613/1641476100.algo-1.json
sagemaker-scikit-learn-2022-01-06-13-32-04-990/profiler-output/system/incremental/2022010
613/1641476160.algo-1.json
sagemaker-scikit-learn-2022-01-06-13-32-04-990/rule-output/ProfilerReport-1641475925/prof
iler-output/profiler-report.html
sagemaker-scikit-learn-2022-01-06-13-32-04-990/rule-output/ProfilerReport-1641475925/prof
iler-output/profiler-report.ipynb
sagemaker-scikit-learn-2022-01-06-13-32-04-990/rule-output/ProfilerReport-1641475925/prof
iler-output/profiler-reports/BatchSize.json
sagemaker-scikit-learn-2022-01-06-13-32-04-990/rule-output/ProfilerReport-1641475925/prof
iler-output/profiler-reports/CPUBottleneck.json
sagemaker-scikit-learn-2022-01-06-13-32-04-990/rule-output/ProfilerReport-1641475925/prof
iler-output/profiler-reports/Dataloader.json
sagemaker-scikit-learn-2022-01-06-13-32-04-990/rule-output/ProfilerReport-1641475925/prof
iler-output/profiler-reports/GPUMemoryIncrease.json
sagemaker-scikit-learn-2022-01-06-13-32-04-990/rule-output/ProfilerReport-1641475925/prof
iler-output/profiler-reports/IOBottleneck.json
sagemaker-scikit-learn-2022-01-06-13-32-04-990/rule-output/ProfilerReport-1641475925/prof
iler-output/profiler-reports/LoadBalancing.json
sagemaker-scikit-learn-2022-01-06-13-32-04-990/rule-output/ProfilerReport-1641475925/prof
iler-output/profiler-reports/LowGPUUtilization.json
sagemaker-scikit-learn-2022-01-06-13-32-04-990/rule-output/ProfilerReport-1641475925/prof
iler-output/profiler-reports/MaxInitializationTime.json
sagemaker-scikit-learn-2022-01-06-13-32-04-990/rule-output/ProfilerReport-1641475925/prof
iler-output/profiler-reports/OverallFrameworkMetrics.json
sagemaker-scikit-learn-2022-01-06-13-32-04-990/rule-output/ProfilerReport-1641475925/prof
iler-output/profiler-reports/OverallSystemUsage.json
sagemaker-scikit-learn-2022-01-06-13-32-04-990/rule-output/ProfilerReport-1641475925/prof
iler-output/profiler-reports/StepOutlier.json
sagemaker-scikit-learn-2022-01-06-13-32-04-990/source/sourcedir.tar.gz
sagemaker-scikit-learn-2022-01-06-13-50-55-389/profiler-output/system/incremental/2022010
613/1641477180.algo-1.json
sagemaker-scikit-learn-2022-01-06-13-50-55-389/profiler-output/system/incremental/2022010
613/1641477240.algo-1.json
sagemaker-scikit-learn-2022-01-06-13-50-55-389/rule-output/ProfilerReport-1641477055/prof
iler-output/profiler-report.html
sagemaker-scikit-learn-2022-01-06-13-50-55-389/rule-output/ProfilerReport-1641477055/prof
iler-output/profiler-report.ipynb
sagemaker-scikit-learn-2022-01-06-13-50-55-389/rule-output/ProfilerReport-1641477055/prof
iler-output/profiler-reports/BatchSize.json
sagemaker-scikit-learn-2022-01-06-13-50-55-389/rule-output/ProfilerReport-1641477055/prof
iler-output/profiler-reports/CPUBottleneck.json
sagemaker-scikit-learn-2022-01-06-13-50-55-389/rule-output/ProfilerReport-1641477055/prof
iler-output/profiler-reports/Dataloader.json
sagemaker-scikit-learn-2022-01-06-13-50-55-389/rule-output/ProfilerReport-1641477055/prof
```

```
iler-output/profiler-reports/GPUMemoryIncrease.json
sagemaker-scikit-learn-2022-01-06-13-50-55-389/rule-output/ProfilerReport-1641477055/prof
iler-output/profiler-reports/IOBottleneck.json
sagemaker-scikit-learn-2022-01-06-13-50-55-389/rule-output/ProfilerReport-1641477055/prof
iler-output/profiler-reports/LoadBalancing.json
sagemaker-scikit-learn-2022-01-06-13-50-55-389/rule-output/ProfilerReport-1641477055/prof
iler-output/profiler-reports/LowGPUUtilization.json
sagemaker-scikit-learn-2022-01-06-13-50-55-389/rule-output/ProfilerReport-1641477055/prof
iler-output/profiler-reports/MaxInitializationTime.json
sagemaker-scikit-learn-2022-01-06-13-50-55-389/rule-output/ProfilerReport-1641477055/prof
iler-output/profiler-reports/OverallFrameworkMetrics.json
sagemaker-scikit-learn-2022-01-06-13-50-55-389/rule-output/ProfilerReport-1641477055/prof
iler-output/profiler-reports/OverallSystemUsage.json
sagemaker-scikit-learn-2022-01-06-13-50-55-389/rule-output/ProfilerReport-1641477055/prof
iler-output/profiler-reports/StepOutlier.json
sagemaker-scikit-learn-2022-01-06-13-50-55-389/source/sourcedir.tar.gz
sagemaker-scikit-learn-2022-01-06-14-22-13-756/debug-output/training_job_end.ts
sagemaker-scikit-learn-2022-01-06-14-22-13-756/output/model.tar.gz
sagemaker-scikit-learn-2022-01-06-14-22-13-756/profiler-output/framework/training_job_end
.ts
sagemaker-scikit-learn-2022-01-06-14-22-13-756/profiler-output/system/incremental/2022010
614/1641479100.algo-1.json
sagemaker-scikit-learn-2022-01-06-14-22-13-756/profiler-output/system/incremental/2022010
614/1641479160.algo-1.json
sagemaker-scikit-learn-2022-01-06-14-22-13-756/profiler-output/system/training_job_end.ts
sagemaker-scikit-learn-2022-01-06-14-22-13-756/rule-output/ProfilerReport-1641478933/prof
iler-output/profiler-report.html
sagemaker-scikit-learn-2022-01-06-14-22-13-756/rule-output/ProfilerReport-1641478933/prof
iler-output/profiler-report.ipynb
sagemaker-scikit-learn-2022-01-06-14-22-13-756/rule-output/ProfilerReport-1641478933/prof
iler-output/profiler-reports/BatchSize.json
sagemaker-scikit-learn-2022-01-06-14-22-13-756/rule-output/ProfilerReport-1641478933/prof
iler-output/profiler-reports/CPUBottleneck.json
sagemaker-scikit-learn-2022-01-06-14-22-13-756/rule-output/ProfilerReport-1641478933/prof
iler-output/profiler-reports/Dataloader.json
sagemaker-scikit-learn-2022-01-06-14-22-13-756/rule-output/ProfilerReport-1641478933/prof
iler-output/profiler-reports/GPUMemoryIncrease.json
sagemaker-scikit-learn-2022-01-06-14-22-13-756/rule-output/ProfilerReport-1641478933/prof
iler-output/profiler-reports/IOBottleneck.json
sagemaker-scikit-learn-2022-01-06-14-22-13-756/rule-output/ProfilerReport-1641478933/prof
iler-output/profiler-reports/LoadBalancing.json
sagemaker-scikit-learn-2022-01-06-14-22-13-756/rule-output/ProfilerReport-1641478933/prof
iler-output/profiler-reports/LowGPUUtilization.json
sagemaker-scikit-learn-2022-01-06-14-22-13-756/rule-output/ProfilerReport-1641478933/prof
iler-output/profiler-reports/MaxInitializationTime.json
sagemaker-scikit-learn-2022-01-06-14-22-13-756/rule-output/ProfilerReport-1641478933/prof
iler-output/profiler-reports/OverallFrameworkMetrics.json
sagemaker-scikit-learn-2022-01-06-14-22-13-756/rule-output/ProfilerReport-1641478933/prof
iler-output/profiler-reports/OverallSystemUsage.json
sagemaker-scikit-learn-2022-01-06-14-22-13-756/rule-output/ProfilerReport-1641478933/prof
iler-output/profiler-reports/StepOutlier.json
sagemaker-scikit-learn-2022-01-06-14-22-13-756/source/sourcedir.tar.gz
Test passed!
```

Modeling

Now that you've uploaded your training data, it's time to define and train a model!

The type of model you create is up to you. For a binary classification task, you can choose to go one of three routes:

- **Use a built-in classification algorithm, like LinearLearner.**
- **Define a custom Scikit-learn classifier, a comparison of models can be found [here](#).**
- **Define a custom PyTorch neural network classifier.**

It will be up to you to test out a variety of models and choose the best one. Your project will be graded on the accuracy of your final model.

EXERCISE: Complete a training script

To implement a custom classifier, you'll need to complete a `train.py` script. You've been given the folders `source_sklearn` and `source_pytorch` which hold starting code for a custom Scikit-learn model and a PyTorch model, respectively. Each directory has a `train.py` training script. To complete this project you only need to complete one of these scripts; the script that is responsible for training your final model.

A typical training script:

- **Loads training data from a specified directory**
- **Parses any training & model hyperparameters (ex. nodes in a neural network, training epochs, etc.)**
- **Instantiates a model of your design, with any specified hyperparams**
- **Trains that model**
- **Finally, saves the model so that it can be hosted/deployed, later**

Defining and training a model

Much of the training script code is provided for you. Almost all of your work will be done in the `if __name__ == '__main__':` section. To complete a `train.py` file, you will:

1. **Import any extra libraries you need**
2. **Define any additional model training hyperparameters using `parser.add_argument`**
3. **Define a model in the `if __name__ == '__main__':` section**
4. **Train the model in that same section**

Below, you can use `!pygmentize` to display an existing `train.py` file. Read through the code; all of your tasks are marked with `TODO` comments.

Note: If you choose to create a custom PyTorch model, you will be responsible for defining the model in the `model.py` file, and a `predict.py` file is provided. If you choose to use Scikit-learn, you only need a `train.py` file; you may import a classifier from the `sklearn` library.

In [5]:

```
# directory can be changed to: source_sklearn or source_pytorch
!pygmentize source_sklearn/train.py
```

```
from __future__ import print_function

import argparse
import os
import pandas as pd

# sklearn.externals.joblib is deprecated in 0.21 and will be removed in 0.23.
# from sklearn.externals import joblib
# Import joblib package directly
import joblib

## TODO: Import any additional libraries you need to define a model
#from sklearn.neighbors import KNeighborsClassifier
from sklearn.gaussian_process import GaussianProcessClassifier
from sklearn.gaussian_process.kernels import RBF

# Provided model load function
def model_fn(model_dir):
    """Load model from the model_dir. This is the same model that is saved
    in the main if statement.
    """
    print("Loading model.")

    # load using joblib
    model = joblib.load(os.path.join(model_dir, "model.joblib"))
    print("Done loading model.")
```

```
    return model


## TODO: Complete the main code
if __name__ == '__main__':

    # All of the model parameters and training parameters are sent as arguments
    # when this script is executed, during a training job

    # Here we set up an argument parser to easily access the parameters
    parser = argparse.ArgumentParser()

    # SageMaker parameters, like the directories for training data and saving models; set
automatically
    # Do not need to change
    parser.add_argument('--output-data-dir', type=str, default=os.environ['SM_OUTPUT_DATA
_DIR'])
    parser.add_argument('--model-dir', type=str, default=os.environ['SM_MODEL_DIR'])
    #parser.add_argument('--data-dir', type=str, default=os.environ['SM_CHANNEL_TRAIN'])
#original
    parser.add_argument('--data-dir', type=str, default=os.environ['SM_CHANNEL_TRAINING']
) #fix from comments

    ## TODO: Add any additional arguments that you will need to pass into your model
    #parser.add_argument('--n_clusters', type=int, default=5)

    # args holds all passed-in arguments
    args = parser.parse_args()

    # Read in csv training file
    training_dir = args.data_dir
    train_data = pd.read_csv(os.path.join(training_dir, "train.csv"), header=None, names=
None)

    # Labels are in the first column
    train_y = train_data.iloc[:,0]
    train_x = train_data.iloc[:,1:]


    ## --- Your code here --- ##


    ## TODO: Define a model
    #model = KNeighborsClassifier(args.n_clusters) #slightly lower accuracy tahn Gaussian
Process.
    model = GaussianProcessClassifier(1.0 * RBF(1.0))


    ## TODO: Train the model
    model.fit(train_x, train_y)


    ## --- End of your code  --- ##


    # Save the trained model
    joblib.dump(model, os.path.join(args.model_dir, "model.joblib"))
```

Provided code

If you read the code above, you can see that the starter code includes a few things:

- **Model loading (`model_fn`) and saving code**
- **Getting SageMaker's default hyperparameters**
- **Loading the training data by name, `train.csv` and extracting the features and labels, `train_x`, and `train_y`**

If you'd like to read more about model saving with [joblib for sklearn](#) or with [torch.save](#), click on the provided links.

Create an Estimator

When a custom model is constructed in SageMaker, an entry point must be specified. This is the Python file which will be executed when the model is trained; the `train.py` function you specified above. To run a custom training script in SageMaker, construct an estimator, and fill in the appropriate constructor arguments:

- **entry_point: The path to the Python script SageMaker runs for training and prediction.**
- **source_dir: The path to the training script directory `source_sklearn` OR `source_pytorch`.**
- **role: Role ARN, which was specified, above.**
- **train_instance_count: The number of training instances (should be left at 1).**
- **train_instance_type: The type of SageMaker instance for training. Note: Because Scikit-learn does not natively support GPU training, Sagemaker Scikit-learn does not currently support training on GPU instance types.**
- **sagemaker_session: The session used to train on Sagemaker.**
- **hyperparameters (optional): A dictionary `{'name':value, ..}` passed to the train function as hyperparameters.**

Note: For a PyTorch model, there is another optional argument framework_version, which you can set to the latest version of PyTorch, `1.0`.

EXERCISE: Define a Scikit-learn or PyTorch estimator

To import your desired estimator, use one of the following lines:

```
from sagemaker.sklearn.estimator import SKLearn

from sagemaker.pytorch import PyTorch
```

In [6]:

```
# your import and estimator code, here
from sagemaker.sklearn.estimator import SKLearn

estimator = SKLearn(entry_point = "train.py",
                    source_dir = "source_sklearn",
                    role = role,
                    framework_version = '0.23-1',
                    train_instance_count = 1,
                    train_instance_type = 'ml.c4.xlarge',
                    sagemaker_session = session,
                    py_version = 'py3',
                    )
```

```
train_instance_type has been renamed in sagemaker>=2.
See: https://sagemaker.readthedocs.io/en/stable/v2.html for details.
train_instance_count has been renamed in sagemaker>=2.
See: https://sagemaker.readthedocs.io/en/stable/v2.html for details.
train_instance_count has been renamed in sagemaker>=2.
See: https://sagemaker.readthedocs.io/en/stable/v2.html for details.
train_instance_type has been renamed in sagemaker>=2.
See: https://sagemaker.readthedocs.io/en/stable/v2.html for details.
```

EXERCISE: Train the estimator

Train your estimator on the training data stored in S3. This should create a training job that you can monitor in your SageMaker console.

In [7]:

```
%%time
```

```
# Train your estimator on S3 training data
estimator.fit({'training': input_data})
```

```
2022-01-10 15:31:22 Starting - Starting the training job...
2022-01-10 15:31:49 Starting - Launching requested ML instancesProfilerReport-1641828682:
InProgress
.........
2022-01-10 15:33:07 Starting - Preparing the instances for training.........
2022-01-10 15:34:48 Downloading - Downloading input data...
2022-01-10 15:35:10 Training - Downloading the training image..2022-01-10 15:35:32,870 sa
gemaker-containers INFO     Imported framework sagemaker_sklearn_container.training
2022-01-10 15:35:32,872 sagemaker-training-toolkit INFO     No GPUs detected (normal if n
o gpus installed)
2022-01-10 15:35:32,883 sagemaker_sklearn_container.training INFO     Invoking user train
ing script.

2022-01-10 15:35:52 Uploading - Uploading generated training model2022-01-10 15:35:48,750
sagemaker-training-toolkit INFO     No GPUs detected (normal if no gpus installed)
2022-01-10 15:35:48,763 sagemaker-training-toolkit INFO     No GPUs detected (normal if n
o gpus installed)
2022-01-10 15:35:48,777 sagemaker-training-toolkit INFO     No GPUs detected (normal if n
o gpus installed)
2022-01-10 15:35:48,787 sagemaker-training-toolkit INFO     Invoking user script
Training Env:
{
    "additional_framework_parameters": {},
    "channel_input_dirs": {
        "training": "/opt/ml/input/data/training"
    },
    "current_host": "algo-1",
    "framework_module": "sagemaker_sklearn_container.training:main",
    "hosts": [
        "algo-1"
    ],
    "hyperparameters": {},
    "input_config_dir": "/opt/ml/input/config",
    "input_data_config": {
        "training": {
            "TrainingInputMode": "File",
            "S3DistributionType": "FullyReplicated",
            "RecordWrapperType": "None"
        }
    },
    "input_dir": "/opt/ml/input",
    "is_master": true,
    "job_name": "sagemaker-scikit-learn-2022-01-10-15-31-22-446",
    "log_level": 20,
    "master_hostname": "algo-1",
    "model_dir": "/opt/ml/model",
    "module_dir": "s3://sagemaker-us-east-1-854682369069/sagemaker-scikit-learn-2022-01-1
0-15-31-22-446/source/sourcedir.tar.gz",
    "module_name": "train",
    "network_interface_name": "eth0",
    "num_cpus": 4,
    "num_gpus": 0,
    "output_data_dir": "/opt/ml/output/data",
    "output_dir": "/opt/ml/output",
    "output_intermediate_dir": "/opt/ml/output/intermediate",
    "resource_config": {
        "current_host": "algo-1",
        "hosts": [
            "algo-1"
        ],
        "network_interface_name": "eth0"
    },
    "user_entry_point": "train.py"
}
Environment variables:
SM_HOSTS=["algo-1"]
SM_NETWORK_INTERFACE_NAME=eth0
SM_HPS={}
SM_USER_ENTRY_POINT=train.py
```

```
SM_FRAMEWORK_PARAMS={}
SM_RESOURCE_CONFIG={"current_host":"algo-1","hosts":["algo-1"],"network_interface_name":"
eth0"}
SM_INPUT_DATA_CONFIG={"training":{"RecordWrapperType":"None","S3DistributionType":"FullyR
eplicated","TrainingInputMode":"File"}}
SM_OUTPUT_DATA_DIR=/opt/ml/output/data
SM_CHANNELS=["training"]
SM_CURRENT_HOST=algo-1
SM_MODULE_NAME=train
SM_LOG_LEVEL=20
SM_FRAMEWORK_MODULE=sagemaker_sklearn_container.training:main
SM_INPUT_DIR=/opt/ml/input
SM_INPUT_CONFIG_DIR=/opt/ml/input/config
SM_OUTPUT_DIR=/opt/ml/output
SM_NUM_CPUS=4
SM_NUM_GPUS=0
SM_MODEL_DIR=/opt/ml/model
SM_MODULE_DIR=s3://sagemaker-us-east-1-854682369069/sagemaker-scikit-learn-2022-01-10-15-
31-22-446/source/sourcedir.tar.gz
SM_TRAINING_ENV={"additional_framework_parameters":{},"channel_input_dirs":{"training":"/
opt/ml/input/data/training"},"current_host":"algo-1","framework_module":"sagemaker_sklear
n_container.training:main","hosts":["algo-1"],"hyperparameters":{},"input_config_dir":"/o
pt/ml/input/config","input_data_config":{"training":{"RecordWrapperType":"None","S3Distri
butionType":"FullyReplicated","TrainingInputMode":"File"}},"input_dir":"/opt/ml/input","i
s_master":true,"job_name":"sagemaker-scikit-learn-2022-01-10-15-31-22-446","log_level":20
,"master_hostname":"algo-1","model_dir":"/opt/ml/model","module_dir":"s3://sagemaker-us-e
ast-1-854682369069/sagemaker-scikit-learn-2022-01-10-15-31-22-446/source/sourcedir.tar.gz
","module_name":"train","network_interface_name":"eth0","num_cpus":4,"num_gpus":0,"output
_data_dir":"/opt/ml/output/data","output_dir":"/opt/ml/output","output_intermediate_dir":
"/opt/ml/output/intermediate","resource_config":{"current_host":"algo-1","hosts":["algo-1
"],"network_interface_name":"eth0"},"user_entry_point":"train.py"}
SM_USER_ARGS=[]
SM_OUTPUT_INTERMEDIATE_DIR=/opt/ml/output/intermediate
SM_CHANNEL_TRAINING=/opt/ml/input/data/training
PYTHONPATH=/opt/ml/code:/miniconda3/bin:/miniconda3/lib/python37.zip:/miniconda3/lib/pyth
on3.7:/miniconda3/lib/python3.7/lib-dynload:/miniconda3/lib/python3.7/site-packages
Invoking script with the following command:
/miniconda3/bin/python train.py
2022-01-10 15:35:50,472 sagemaker-containers INFO     Reporting training SUCCESS

2022-01-10 15:36:08 Completed - Training job completed
Training seconds: 86
Billable seconds: 86
CPU times: user 675 ms, sys: 74.4 ms, total: 749 ms
Wall time: 5min 15s
```

EXERCISE: Deploy the trained model

After training, deploy your model to create a `predictor`. If you're using a PyTorch model, you'll need to create a trained `PyTorchModel` that accepts the trained `<model>.model_data` as an input parameter and points to the provided `source_pytorch/predict.py` file as an entry point.

To deploy a trained model, you'll use `<model>.deploy`, which takes in two arguments:

- **initial_instance_count: The number of deployed instances (1).**
- **instance_type: The type of SageMaker instance for deployment.**

Note: If you run into an instance error, it may be because you chose the wrong training or deployment instance_type. It may help to refer to your previous exercise code to see which types of instances we used.

In [8]:

```
%%time

# uncomment, if needed
# from sagemaker.pytorch import PyTorchModel


# deploy your model to create a predictor
```

```
predictor = estimator.deploy(initial_instance_count=1, instance_type='ml.m4.xlarge')
```

```
------!CPU times: user 101 ms, sys: 20.8 ms, total: 122 ms
Wall time: 3min 1s
```

Evaluating Your Model

Once your model is deployed, you can see how it performs when applied to our test data.

The provided cell below, reads in the test data, assuming it is stored locally in `data_dir` and named `test.csv`. The labels and features are extracted from the `.csv` file.

In [9]:

```
"""
DON'T MODIFY ANYTHING IN THIS CELL THAT IS BELOW THIS LINE
"""
import os

# read in test data, assuming it is stored locally
test_data = pd.read_csv(os.path.join(data_dir, "test.csv"), header=None, names=None)

# labels are in the first column
test_y = test_data.iloc[:,0]
test_x = test_data.iloc[:,1:]
```

EXERCISE: Determine the accuracy of your model

Use your deployed `predictor` to generate predicted, class labels for the test data. Compare those to the *true* labels, `test_y`, and calculate the accuracy as a value between 0 and 1.0 that indicates the fraction of test data that your model classified correctly. You may use [sklearn.metrics] for this calculation.

To pass this project, your model should get at least 90% test accuracy.

In [10]:

```
# First: generate predicted, class labels
test_y_preds = predictor.predict(test_x)


"""
DON'T MODIFY ANYTHING IN THIS CELL THAT IS BELOW THIS LINE
"""
# test that your model generates the correct number of labels
assert len(test_y_preds)==len(test_y), 'Unexpected number of predictions.'
print('Test passed!')
```

```
Test passed!
```

In [11]:

```
from sklearn.metrics import accuracy_score

# Second: calculate the test accuracy
accuracy = accuracy_score(test_y, test_y_preds)

print(accuracy)


## print out the array of predicted and true labels, if you want
print('\nPredicted class labels: ')
print(test_y_preds)
print('\nTrue class labels: ')
print(test_y.values)
```

```
0.96
```

```
0.96

Predicted class labels:
[1 1 1 1 1 1 1 0 0 0 0 0 1 1 1 1 1 1 0 1 0 1 1 0 0]

True class labels:
[1 1 1 1 1 1 1 0 0 0 0 0 1 1 1 1 1 1 0 1 0 1 1 0 0]
```

Question 1: How many false positives and false negatives did your model produce, if any? And why do you think this is?

Answer:

The nearest neighbors algorithm produced two false positive cases and no false negative cases. Which means that there are two positive datapoints that lie on the "negative side" of the decision boundary betwene the two clusters the algorithm created. The Guassian Process algorithm produced one false positive and no false negative cases, which is the same one as in the nearest neighbors. Most likely this datapoint lies very close to the actual positive values and is therefore misclassified.

Question 2: How did you decide on the type of model to use?

Answer: I decided to try the two sklearn methods that scored an accuracy higher than 0.9 for all datatypes in the article provided. These are the Nearest Neighbors and Gaussian Process algorithms. These are the results:

1. **The nearest neighbors, using 2 clusters, had an accuracy score of 0.92**
2. **The Gaussian Process algorithm had an accuracy score of 0.96.**

Because the Gaussian Process had the highest accuracy for our data, I decided to use this type of model.

EXERCISE: Clean up Resources

After you're done evaluating your model, delete your model endpoint. You can do this with a call to `.delete_endpoint()`. You need to show, in this notebook, that the endpoint was deleted. Any other resources, you may delete from the AWS console, and you will find more instructions on cleaning up all your resources, below.

In [12]:

```
# uncomment and fill in the line below!
predictor.delete_endpoint()
```

Deleting S3 bucket

When you are *completely* done with training and testing models, you can also delete your entire S3 bucket. If you do this before you are done training your model, you'll have to recreate your S3 bucket and upload your training data again.

In [13]:

```
# deleting bucket, uncomment lines below

# bucket_to_delete = boto3.resource('s3').Bucket(bucket)
# bucket_to_delete.objects.all().delete()
```

Deleting all your models and instances

When you are *completely* done with this project and do not ever want to revisit this notebook, you can choose to delete all of your SageMaker notebook instances and models by following these instructions. Before you delete this notebook instance, I recommend at least downloading a copy and saving it, locally.

Further Directions

There are many ways to improve or add on to this project to expand your learning or make this more of a unique project for you. A few ideas are listed below:

- **Train a classifier to predict the *category* (1-3) of plagiarism and not just plagiarized (1) or not (0).**
- **Utilize a different and larger dataset to see if this model can be extended to other types of plagiarism.**
- **Use language or character-level analysis to find different (and more) similarity features.**
- **Write a complete pipeline function that accepts a source text and submitted text file, and classifies the submitted text as plagiarized or not.**
- **Use API Gateway and a lambda function to deploy your model to a web application.**

These are all just options for extending your work. If you've completed all the exercises in this notebook, you've completed a real-world application, and can proceed to submit your project. Great job!